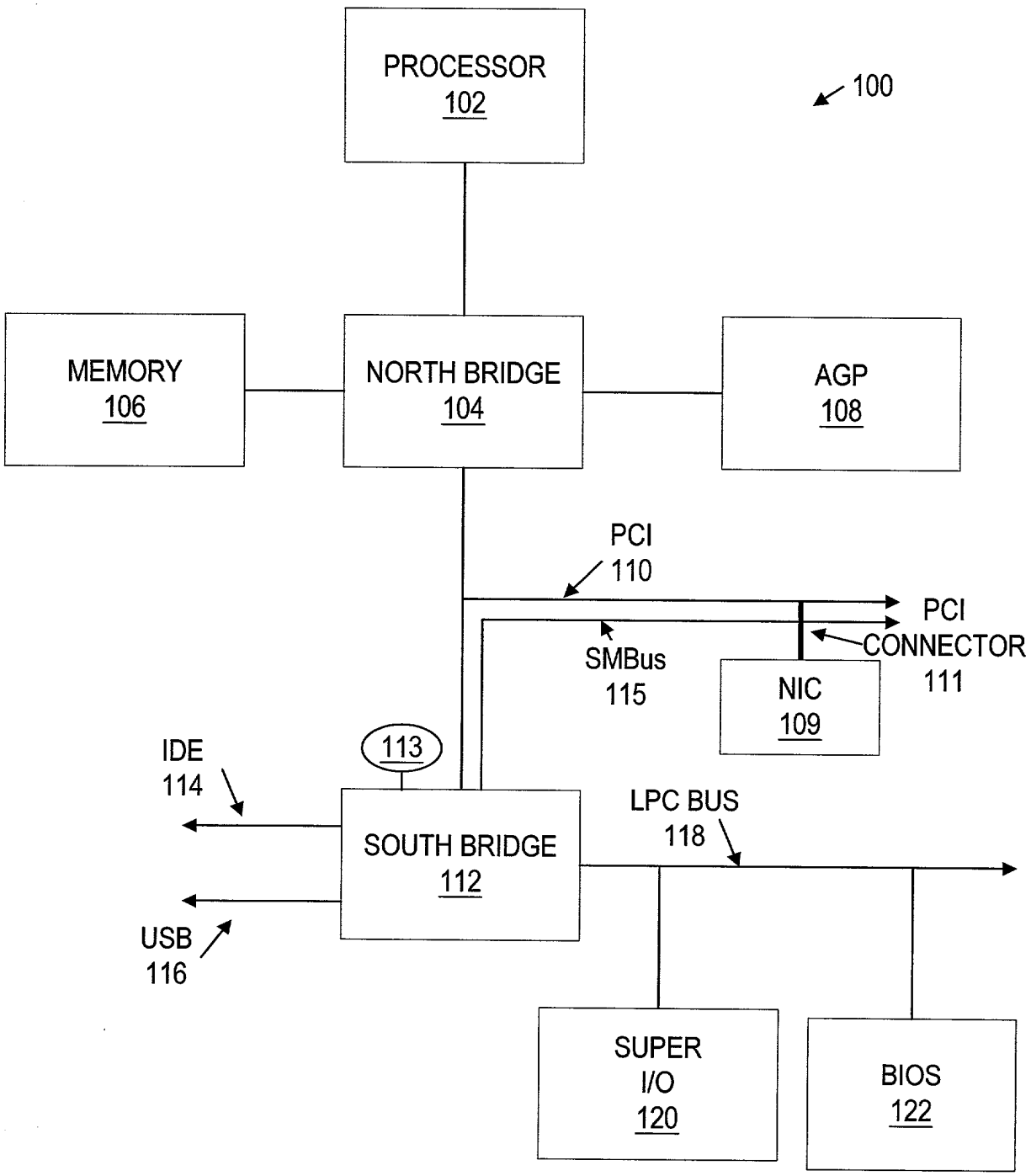


Fig. 1A
(Prior Art)

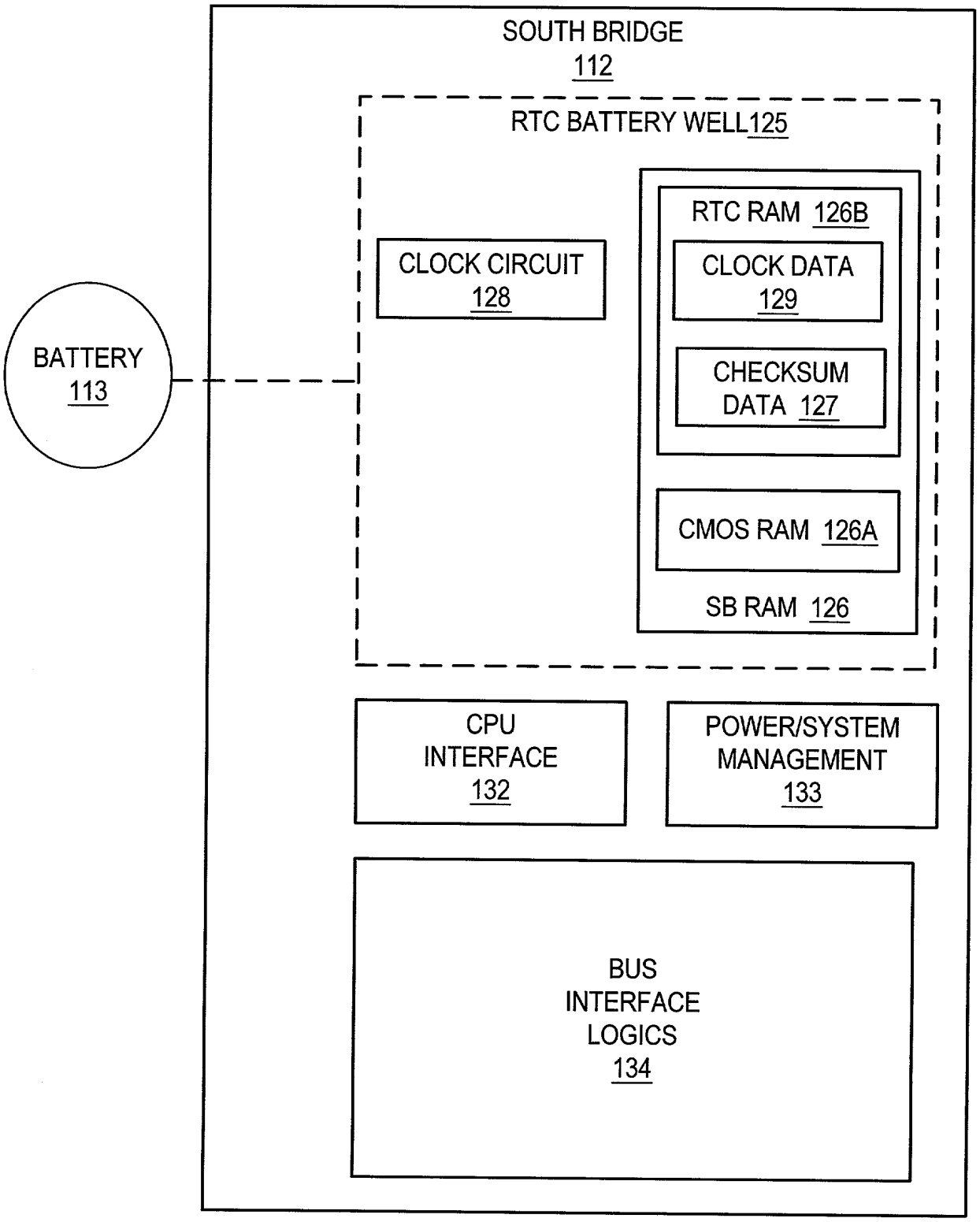


Fig. 1B
(Prior Art)

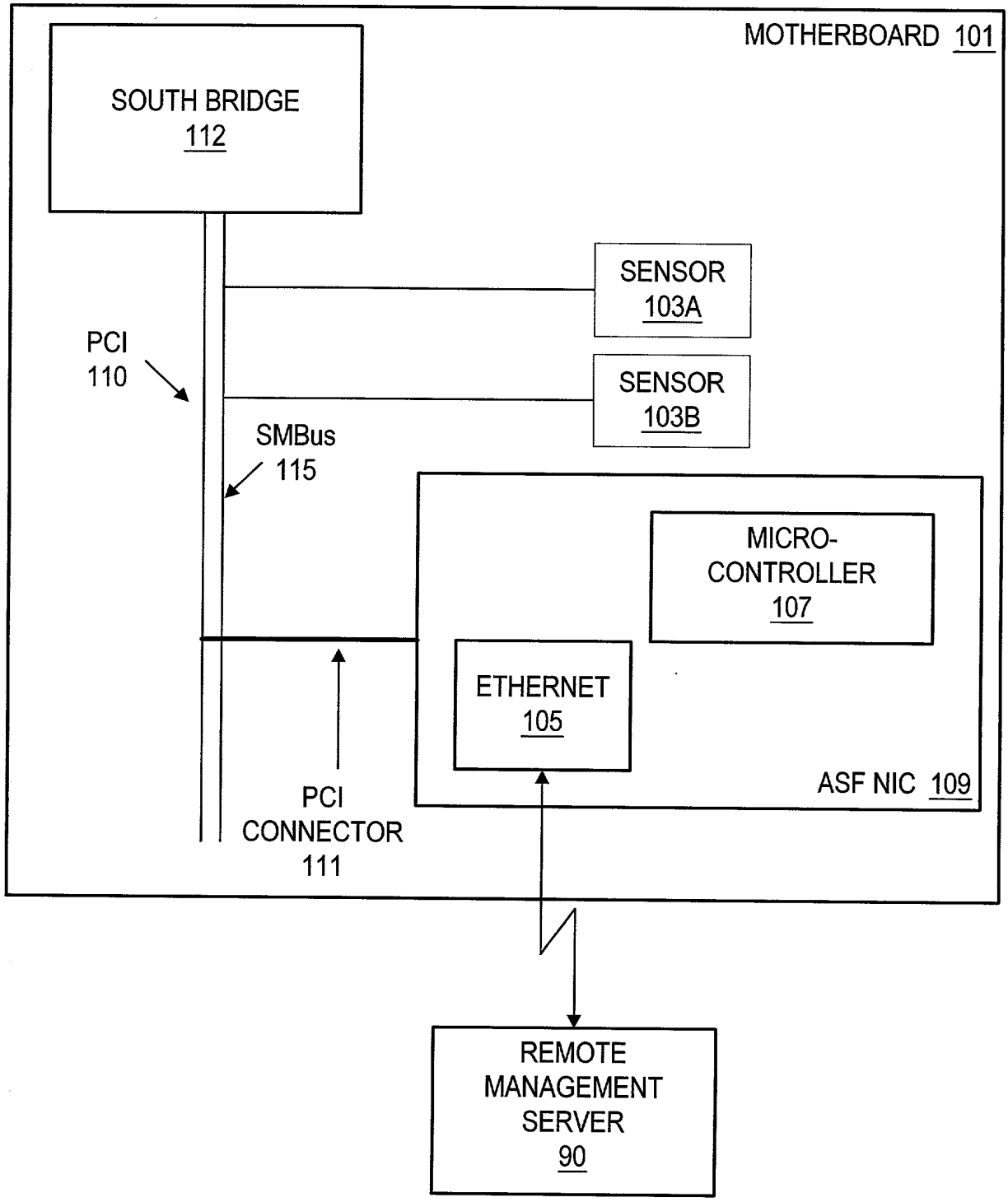


**Fig. 1C**
**(Prior Art)**

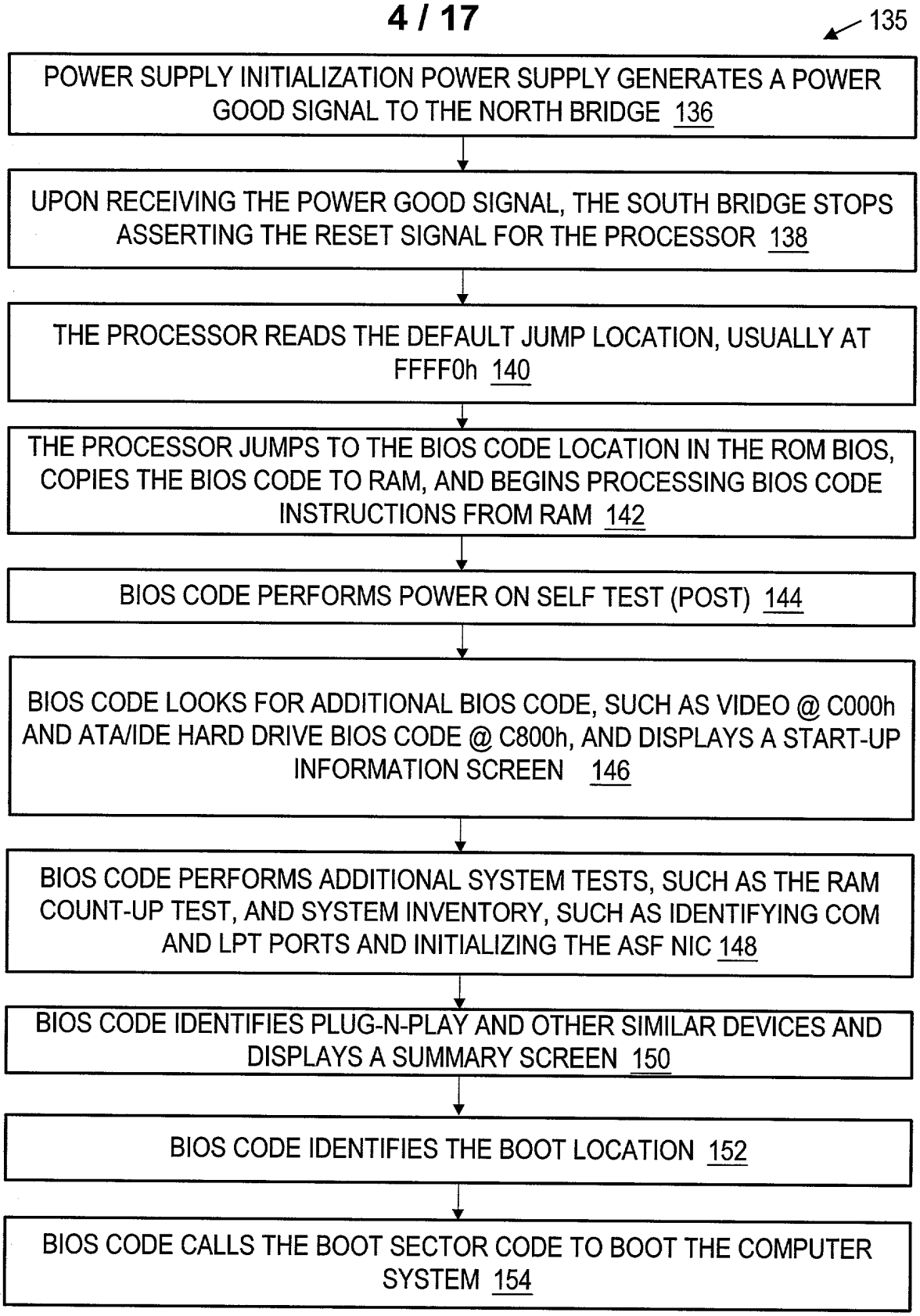


**Fig. 2**
**(Prior Art)**

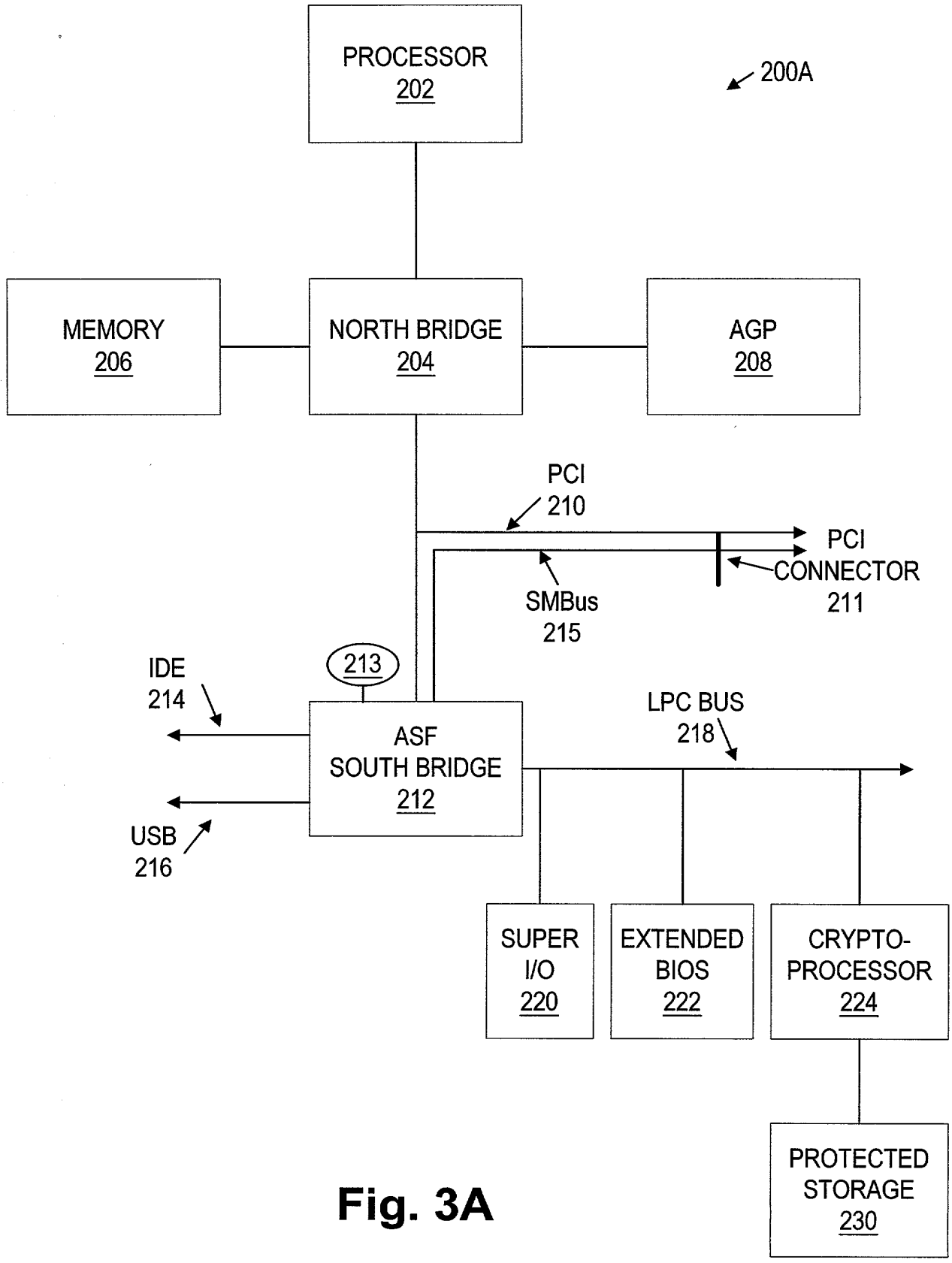
PROCESSOR
202
200A
MEMORY
206
NORTH BRIDGE
204
AGP
208
PCI
210
PCI
CONNECTOR
211
SMBus
215
213
IDE
214
ASF
SOUTH BRIDGE
212
LPC BUS
218
USB
216
SUPER
I/O
220
EXTENDED
BIOS
222
CRYPTO-
PROCESSOR
224
PROTECTED
STORAGE
230

Fig. 3A

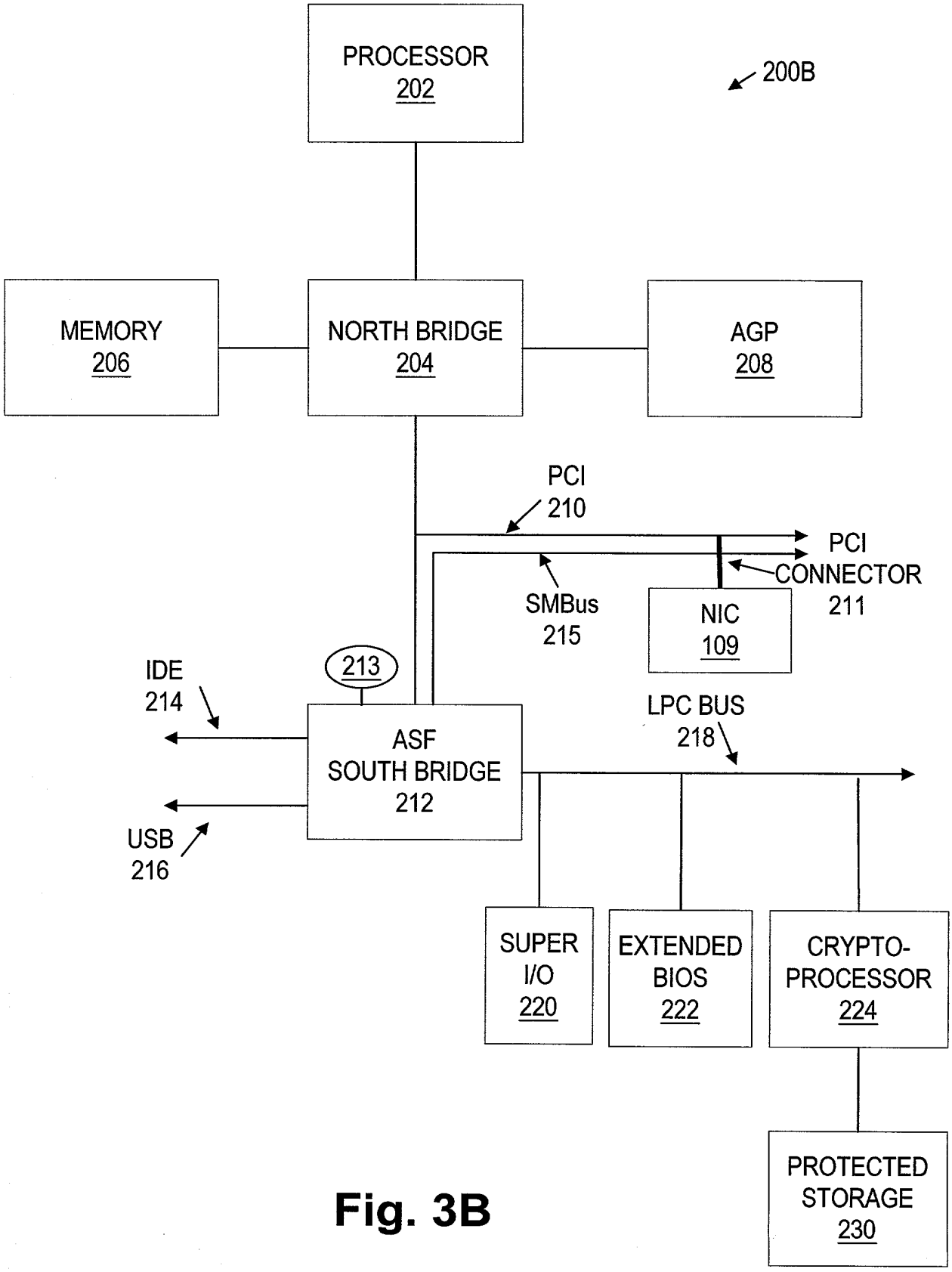
PROCESSOR
202
200B
MEMORY
206
NORTH BRIDGE
204
AGP
208
PCI
210
PCI
CONNECTOR
211
SMBus
215
NIC
109
213
IDE
214
LPC BUS
218
ASF
SOUTH BRIDGE
212
USB
216
SUPER
I/O
220
EXTENDED
BIOS
222
CRYPTO-
PROCESSOR
224
PROTECTED
STORAGE
230

Fig. 3B

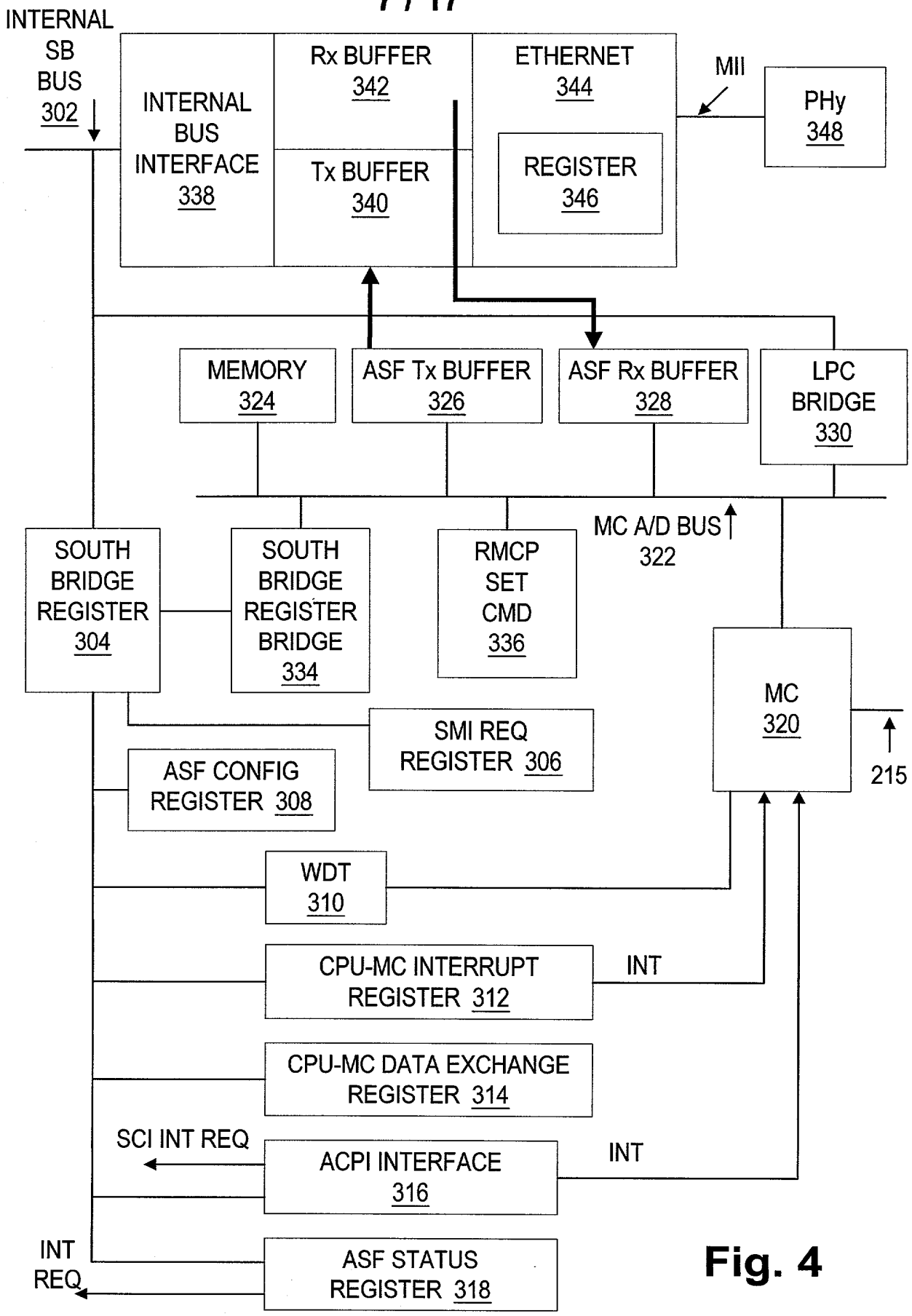


Fig. 4

20070205 0844

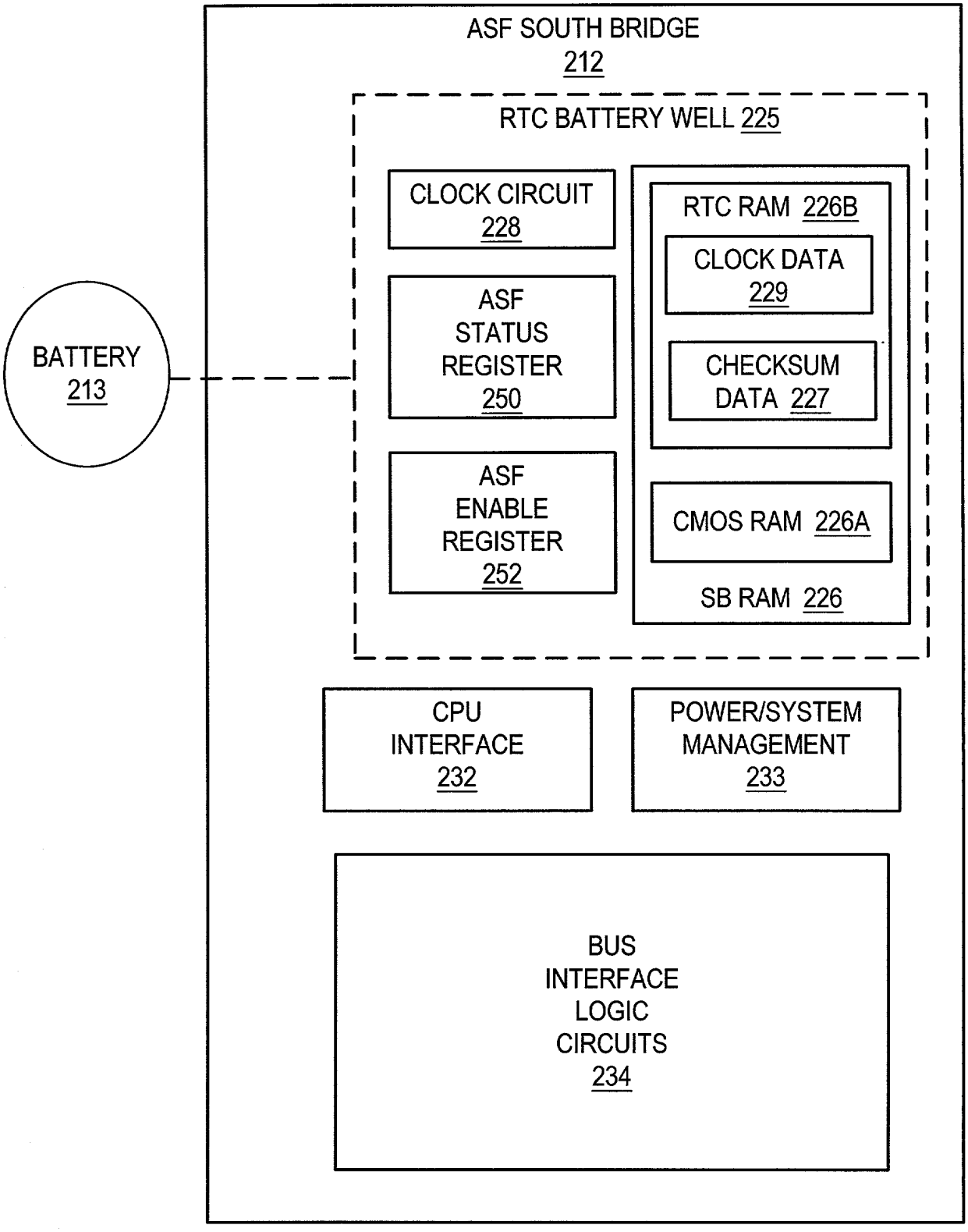
ASF SOUTH BRIDGE
212
RTC BATTERY WELL 225
CLOCK CIRCUIT
228
ASF
STATUS
REGISTER
250
ASF
ENABLE
REGISTER
252
RTC RAM 226B
CLOCK DATA
229
CHECKSUM
DATA 227
CMOS RAM 226A
SB RAM 226
BATTERY
213
CPU
INTERFACE
232
POWER/SYSTEM
MANAGEMENT
233
BUS
INTERFACE
LOGIC
CIRCUITS
234

Fig. 5

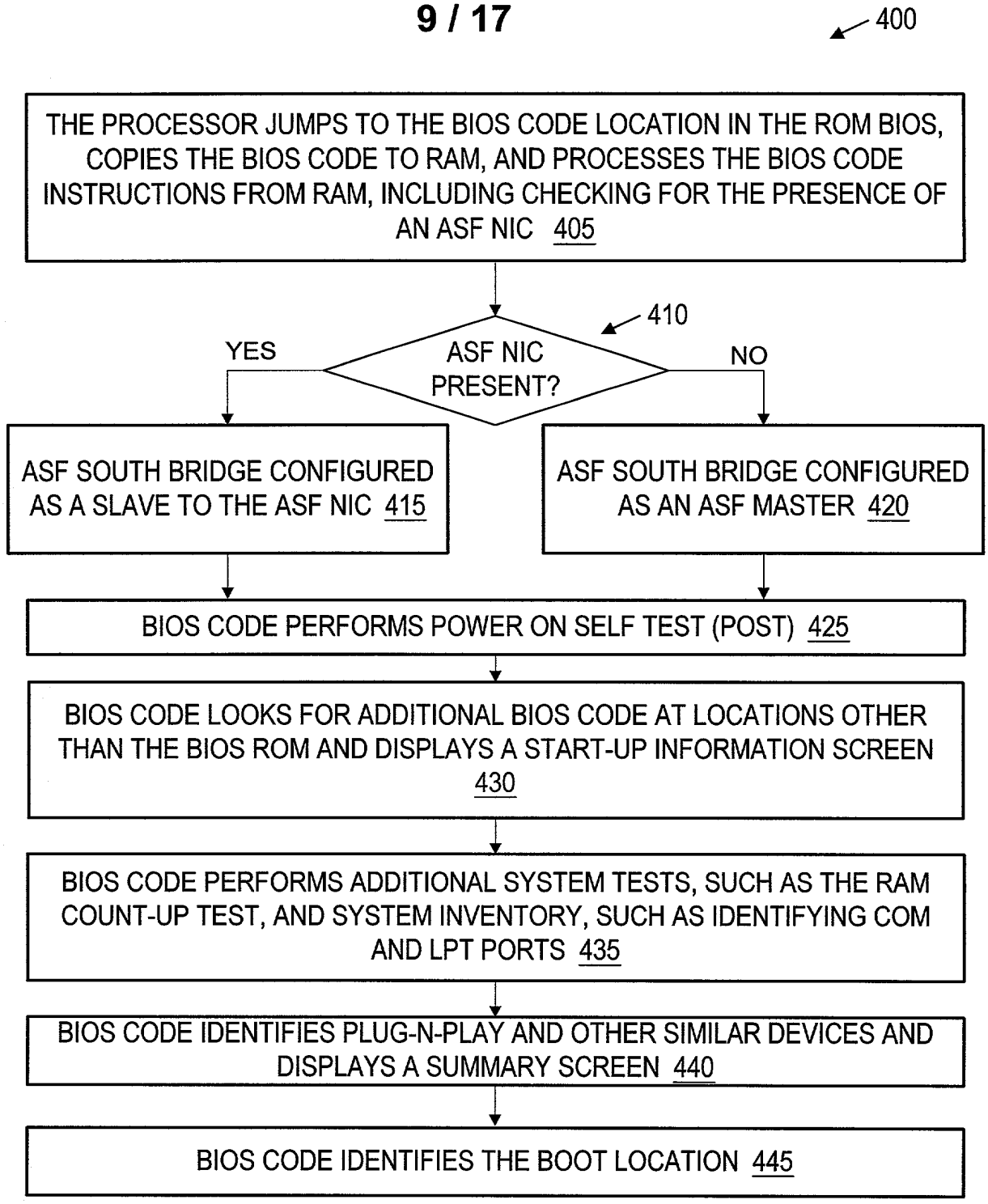
400
THE PROCESSOR JUMPS TO THE BIOS CODE LOCATION IN THE ROM BIOS, COPIES THE BIOS CODE TO RAM, AND PROCESSES THE BIOS CODE INSTRUCTIONS FROM RAM, INCLUDING CHECKING FOR THE PRESENCE OF AN ASF NIC 405
410
ASF NIC PRESENT?
YES
NO
ASF SOUTH BRIDGE CONFIGURED AS A SLAVE TO THE ASF NIC 415
ASF SOUTH BRIDGE CONFIGURED AS AN ASF MASTER 420
BIOS CODE PERFORMS POWER ON SELF TEST (POST) 425
BIOS CODE LOOKS FOR ADDITIONAL BIOS CODE AT LOCATIONS OTHER THAN THE BIOS ROM AND DISPLAYS A START-UP INFORMATION SCREEN 430
BIOS CODE PERFORMS ADDITIONAL SYSTEM TESTS, SUCH AS THE RAM COUNT-UP TEST, AND SYSTEM INVENTORY, SUCH AS IDENTIFYING COM AND LPT PORTS 435
BIOS CODE IDENTIFIES PLUG-N-PLAY AND OTHER SIMILAR DEVICES AND DISPLAYS A SUMMARY SCREEN 440
BIOS CODE IDENTIFIES THE BOOT LOCATION 445

Fig. 6

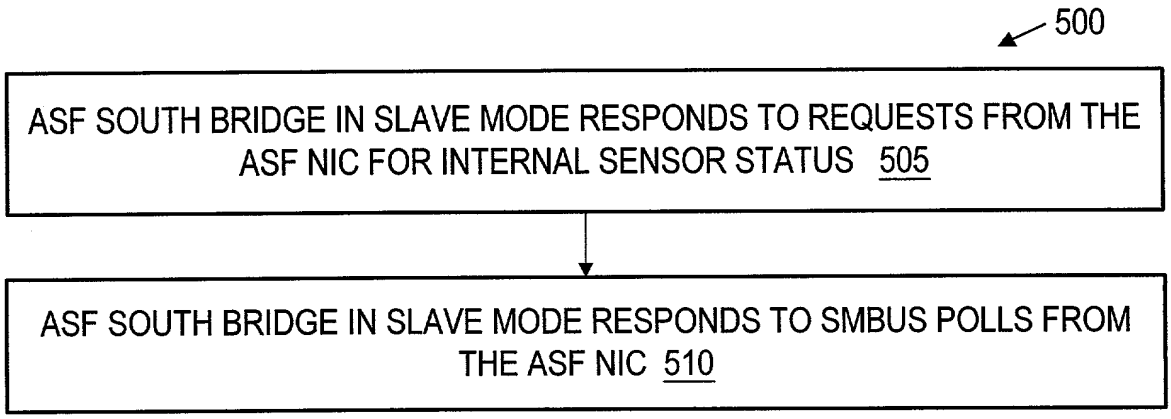


Fig. 7A

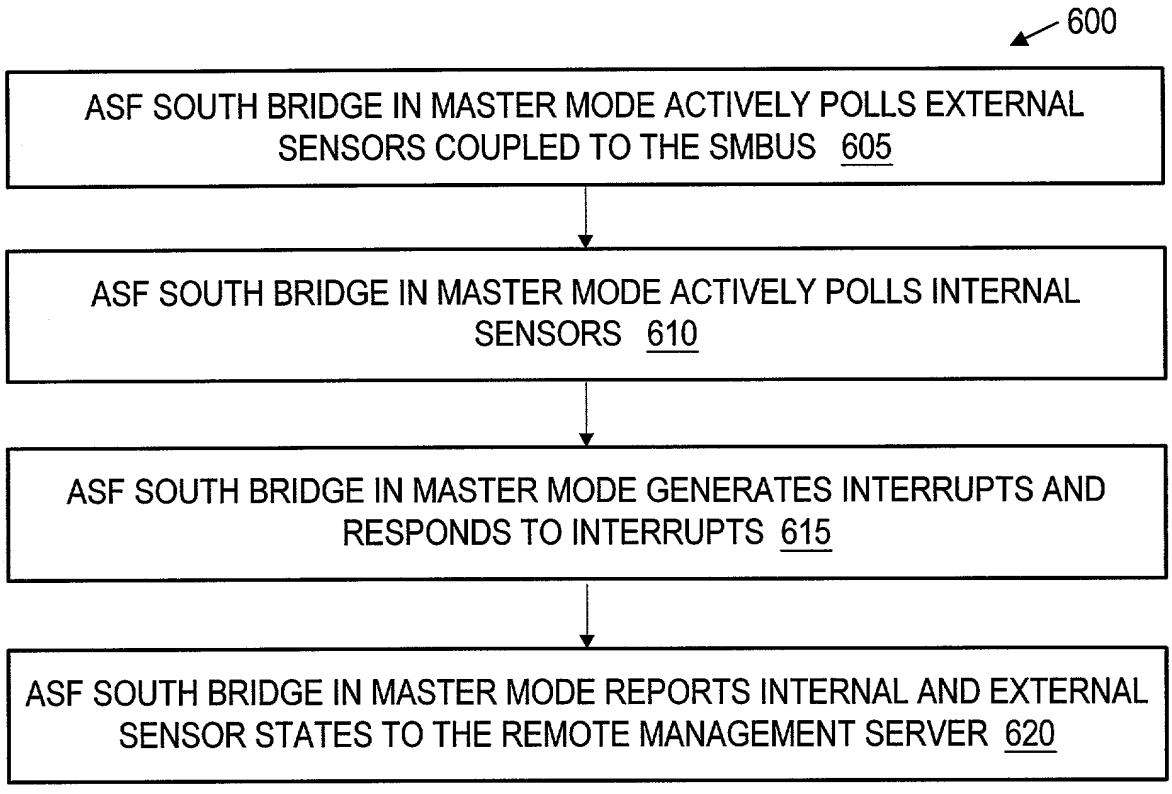


Fig. 7B

20070207.0204

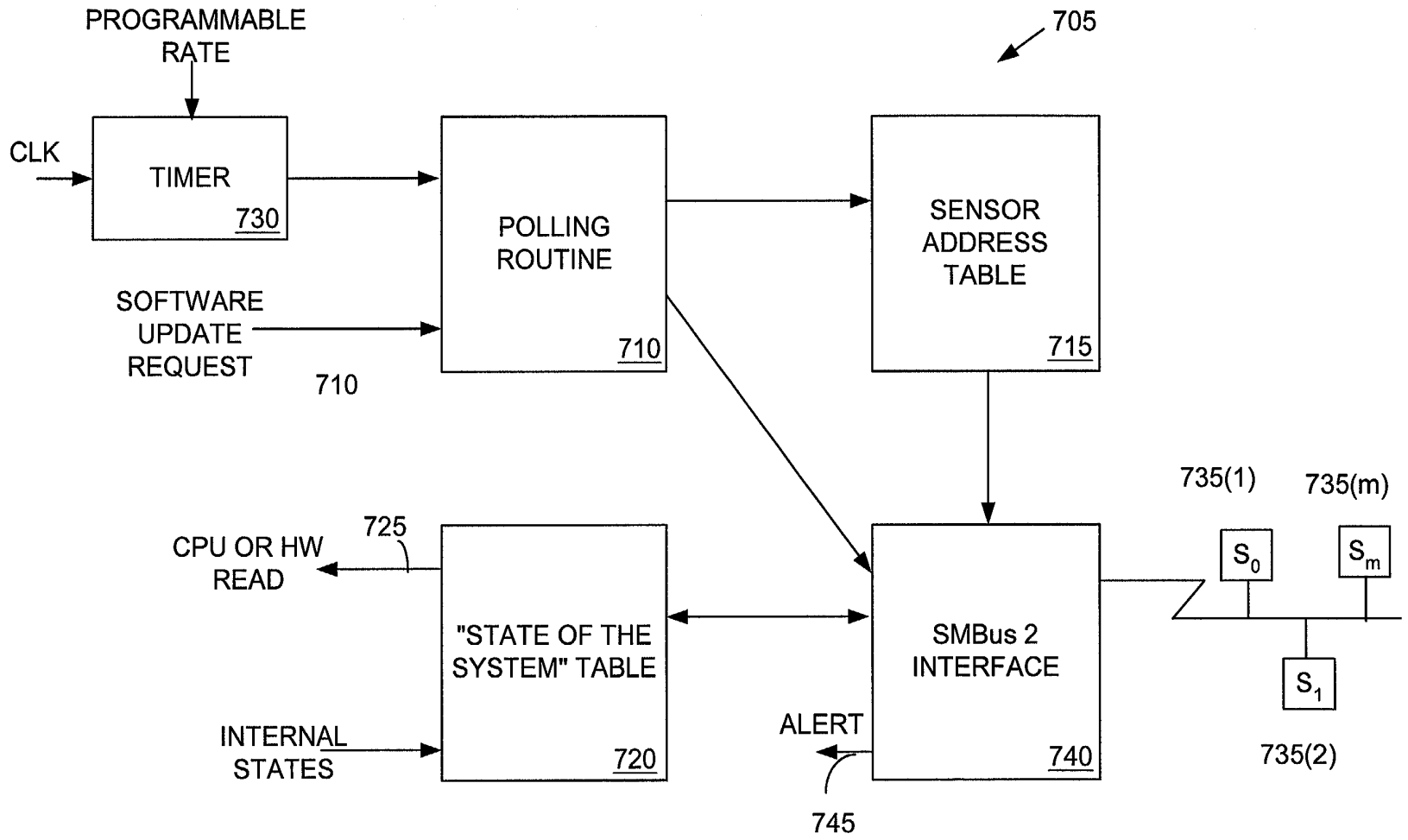
PROGRAMMABLE RATE
705
CLK
TIMER
730
POLLING ROUTINE
710
SENSOR ADDRESS TABLE
715
SOFTWARE UPDATE REQUEST
710
735(1)
735(m)
725
CPU OR HW READ
"STATE OF THE SYSTEM" TABLE
720
SMBus 2 INTERFACE
740
$S_0$
$S_m$
$S_1$
INTERNAL STATES
ALERT
745
735(2)

Fig. 7C

| Offset | Name | Description |
|---|---|---|
| 00h | Alert, Device Address (n ranges from 1 to Number of Alerts) | Contains the SMBus address to which the SMBus command is written, in the following format:<br>**Bits(s) Description**<br>7:1 Contains the SMBus address of the alert-capable device.<br>0 Identifies where (1) or not (0) the alerting condition is based on an exact match. |
| 01h | Alert, Command | Contains the SMBus Command or register value used to obtain the data associated with the alert. |
| 02h | Alert, Data Mask | The data-mask to be logically AND'd with to the returned sensor data to isolate significant bits associated with this alert. |
| 03h | Alert, Compare Value | The value to compare to the masked sensor data, if an exact match is specified in the Device Address field. If a match results, the alert associated with this condition should be sent by the alerting device. |
| 04h | Alert, Event Sensor Type | The value to be set into the alert's Event Sensor Type field. |
| 05h | Alert, Event Type | The value to be set into the alert's Event Type field. |
| 06h | Alert, Event Offset | The value to be set into the alert's Event Offset field. |
| 07h | Alert, Event Source Type | The value to be set into the alert's Event Source Type field. |
| 08h | Alert, Event Severity | The value to be set into the alert's Event Severity field. |
| 09h | Alert, Sensor Number | The value to be set into the alert's Sensor Number field. |
| 0Ah | Alert, Entity | The value to be set into the alert's Entity field. |
| 0Bh | Alert, Entity Instance | The value to be set into the alert's Entity Instance field. |

Fig. 7D

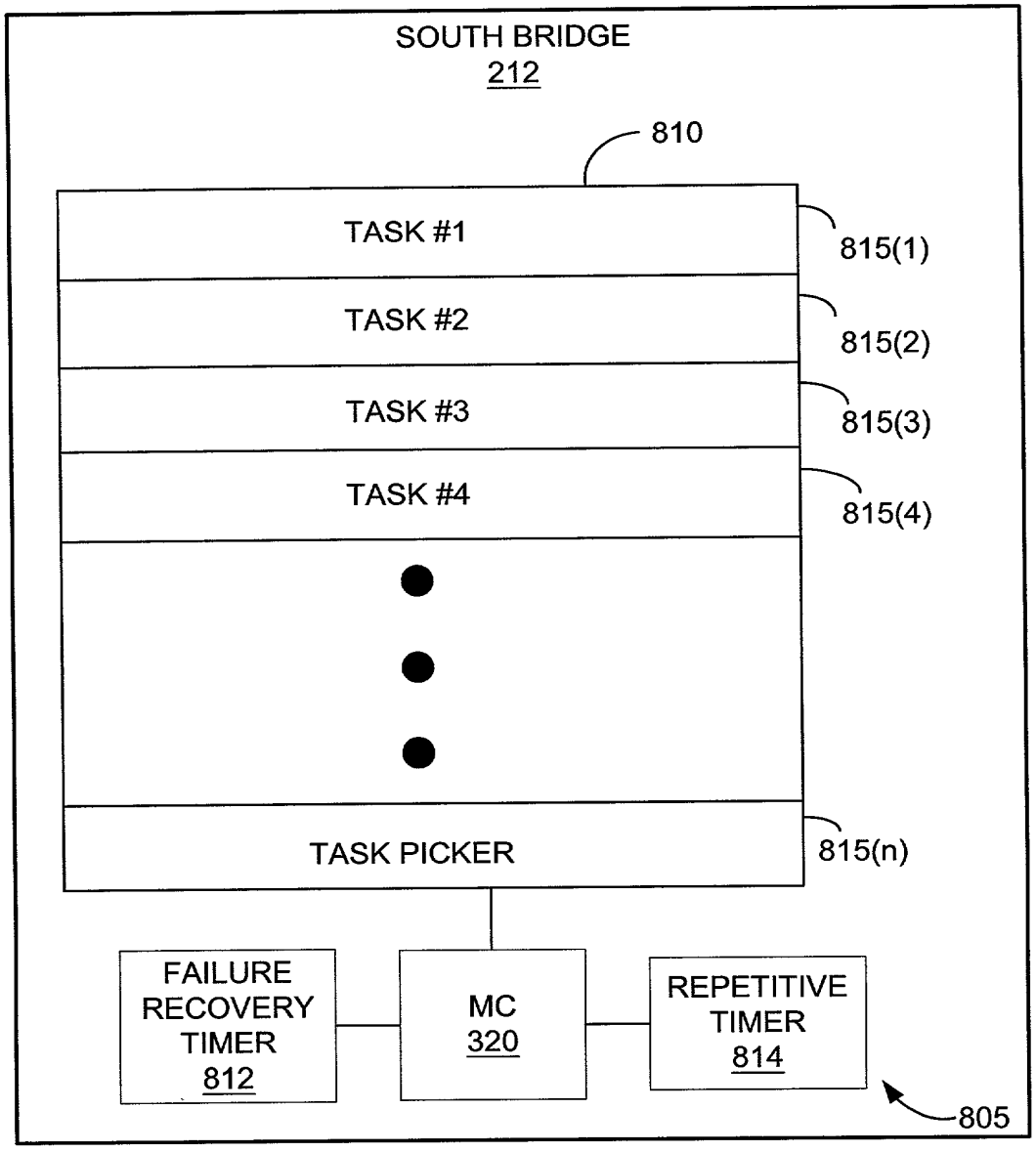
SOUTH BRIDGE
212
810
TASK #1
815(1)
TASK #2
815(2)
TASK #3
815(3)
TASK #4
815(4)
TASK PICKER
815(n)
FAILURE RECOVERY TIMER
812
MC
320
REPETITIVE TIMER
814
805

Fig. 8

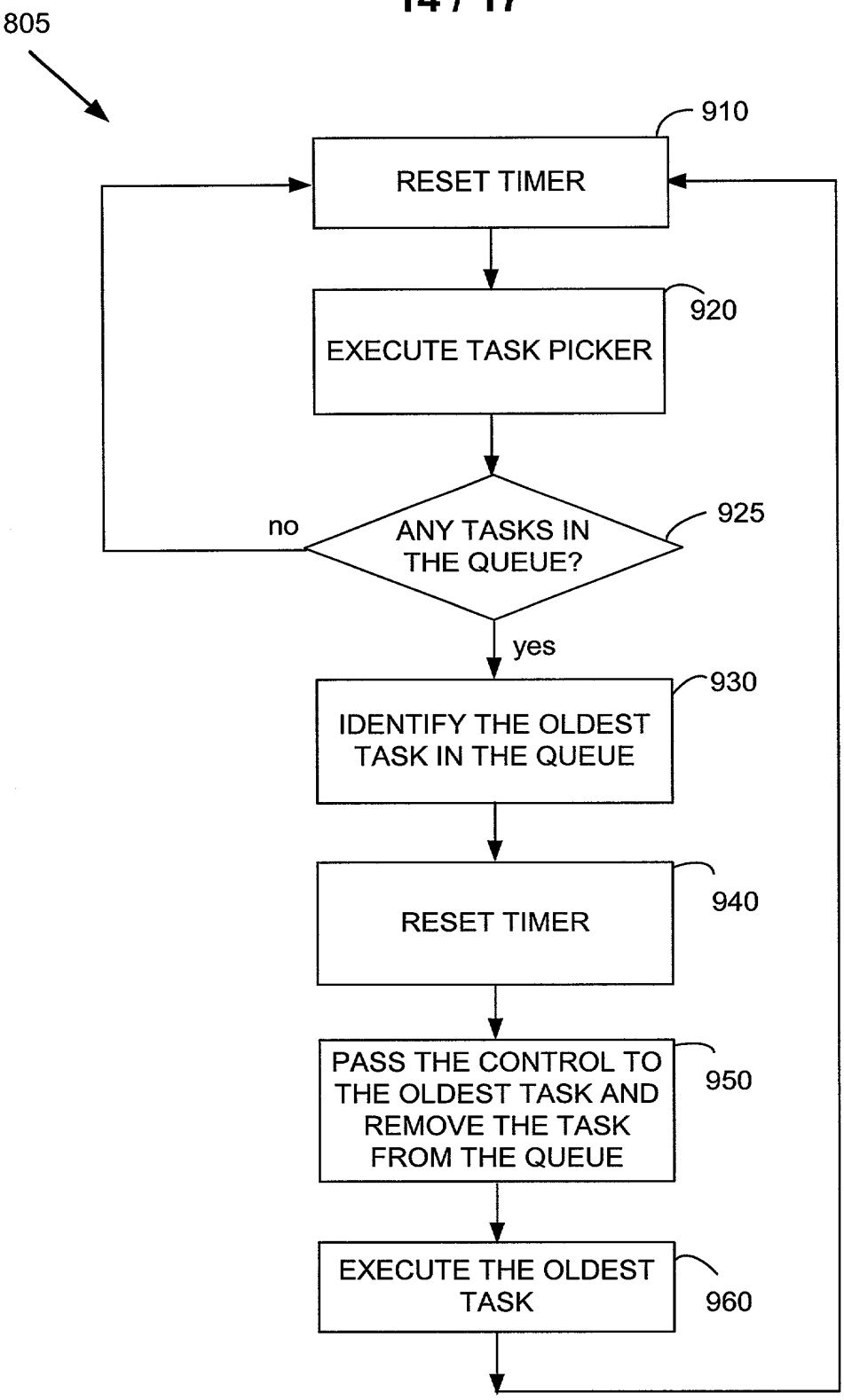
805
910
RESET TIMER
920
EXECUTE TASK PICKER
925
ANY TASKS IN THE QUEUE?
no
yes
930
IDENTIFY THE OLDEST TASK IN THE QUEUE
940
RESET TIMER
950
PASS THE CONTROL TO THE OLDEST TASK AND REMOVE THE TASK FROM THE QUEUE
960
EXECUTE THE OLDEST TASK

Fig. 9

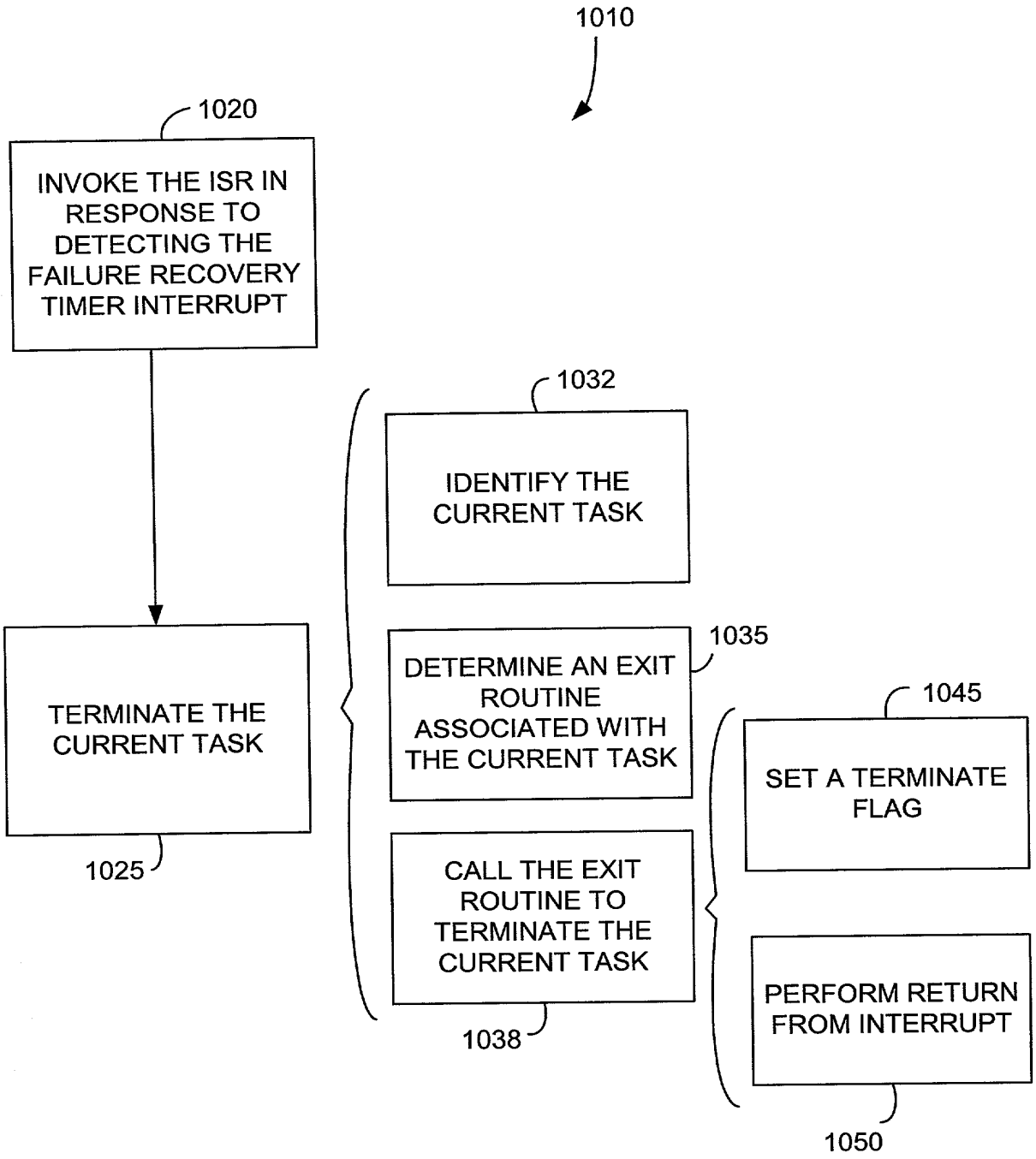
1010
1020
INVOKE THE ISR IN RESPONSE TO DETECTING THE FAILURE RECOVERY TIMER INTERRUPT
TERMINATE THE CURRENT TASK
1025
1032
IDENTIFY THE CURRENT TASK
1035
DETERMINE AN EXIT ROUTINE ASSOCIATED WITH THE CURRENT TASK
CALL THE EXIT ROUTINE TO TERMINATE THE CURRENT TASK
1038
1045
SET A TERMINATE FLAG
PERFORM RETURN FROM INTERRUPT
1050

Fig. 10

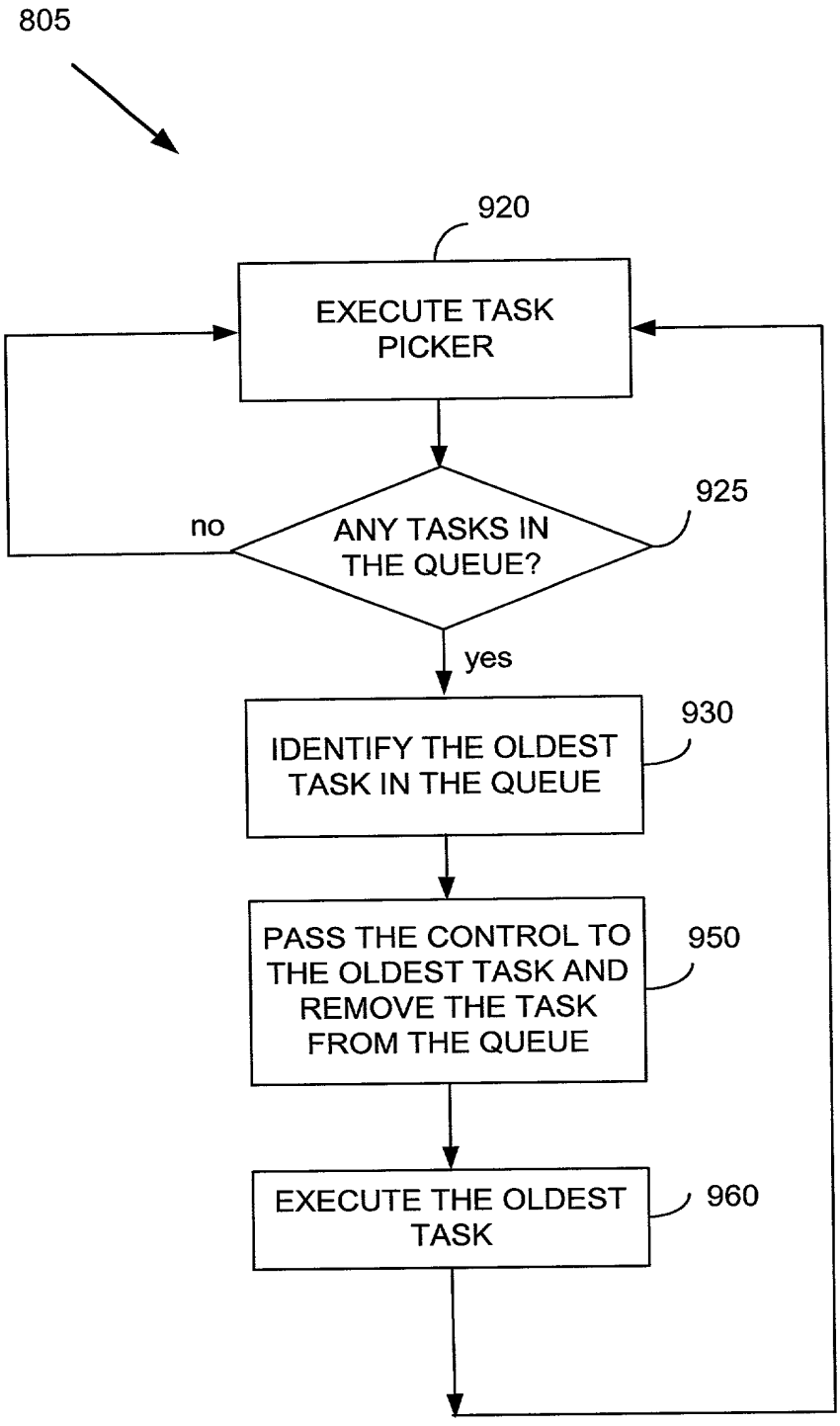
805
920
EXECUTE TASK PICKER
925
ANY TASKS IN THE QUEUE?
no
yes
930
IDENTIFY THE OLDEST TASK IN THE QUEUE
950
PASS THE CONTROL TO THE OLDEST TASK AND REMOVE THE TASK FROM THE QUEUE
960
EXECUTE THE OLDEST TASK

Fig. 11

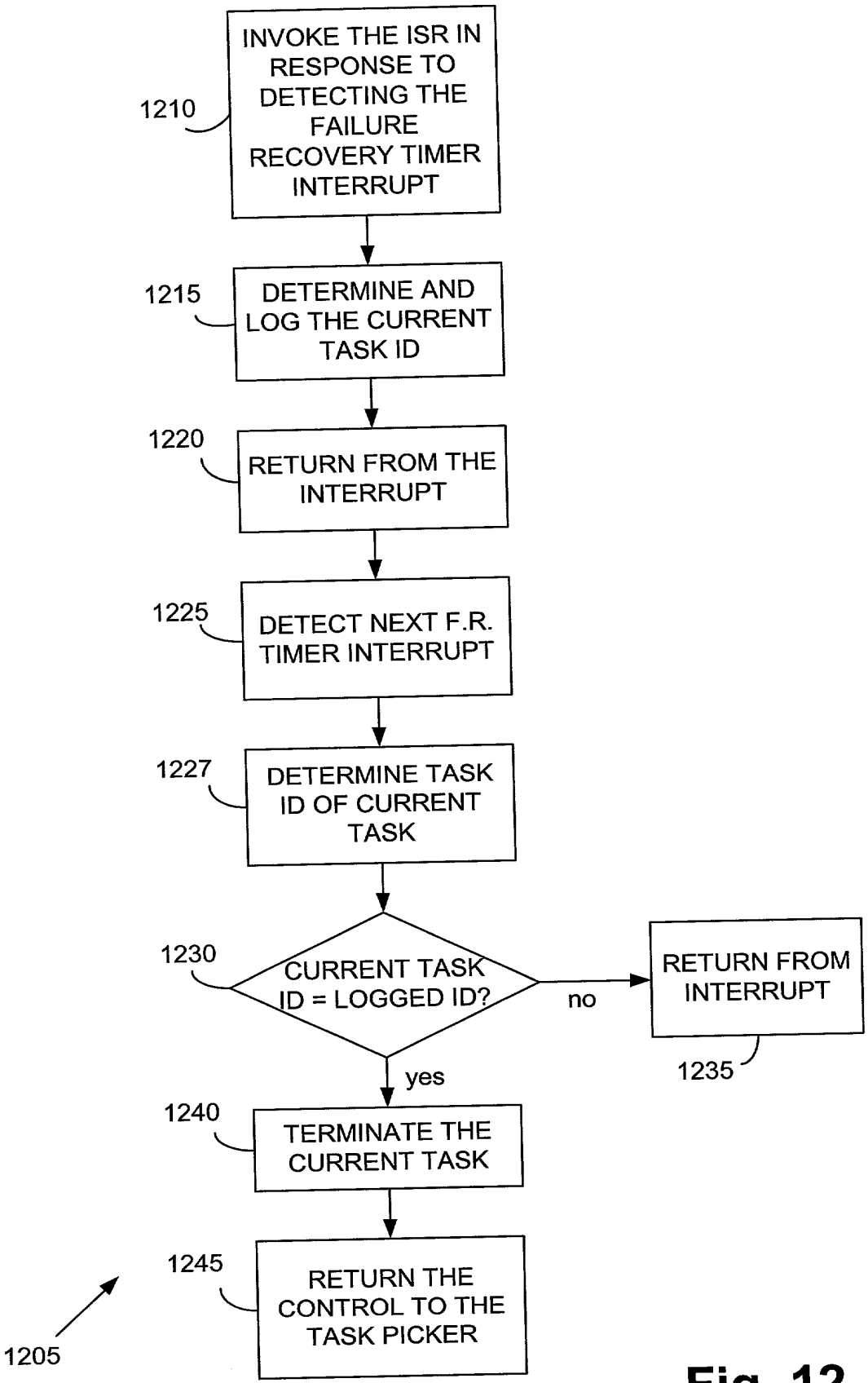
INVOKE THE ISR IN RESPONSE TO DETECTING THE FAILURE RECOVERY TIMER INTERRUPT
1210
DETERMINE AND LOG THE CURRENT TASK ID
1215
RETURN FROM THE INTERRUPT
1220
DETECT NEXT F.R. TIMER INTERRUPT
1225
DETERMINE TASK ID OF CURRENT TASK
1227
CURRENT TASK ID = LOGGED ID?
1230
no
RETURN FROM INTERRUPT
1235
yes
TERMINATE THE CURRENT TASK
1240
RETURN THE CONTROL TO THE TASK PICKER
1245
1205

Fig. 12